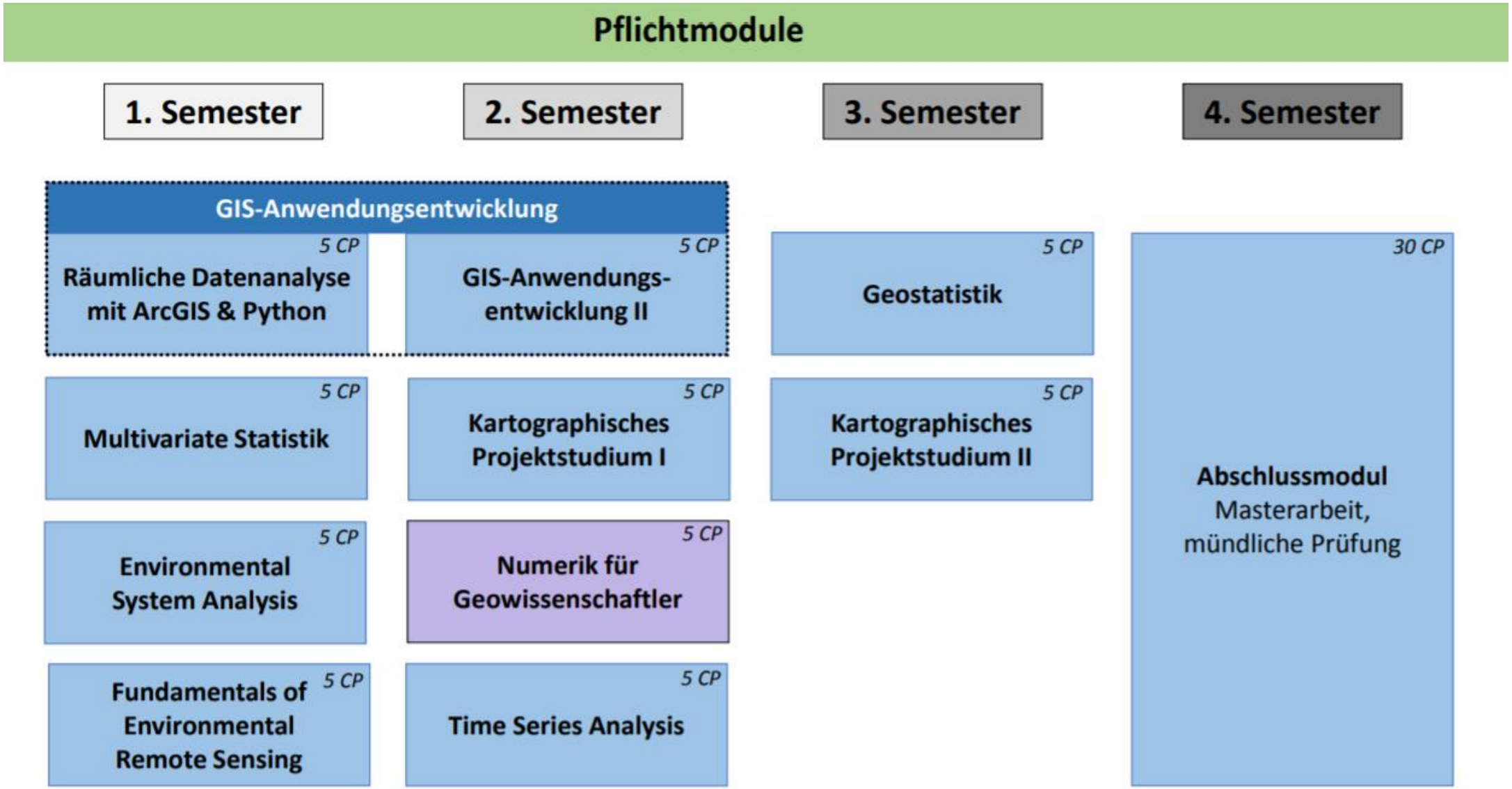
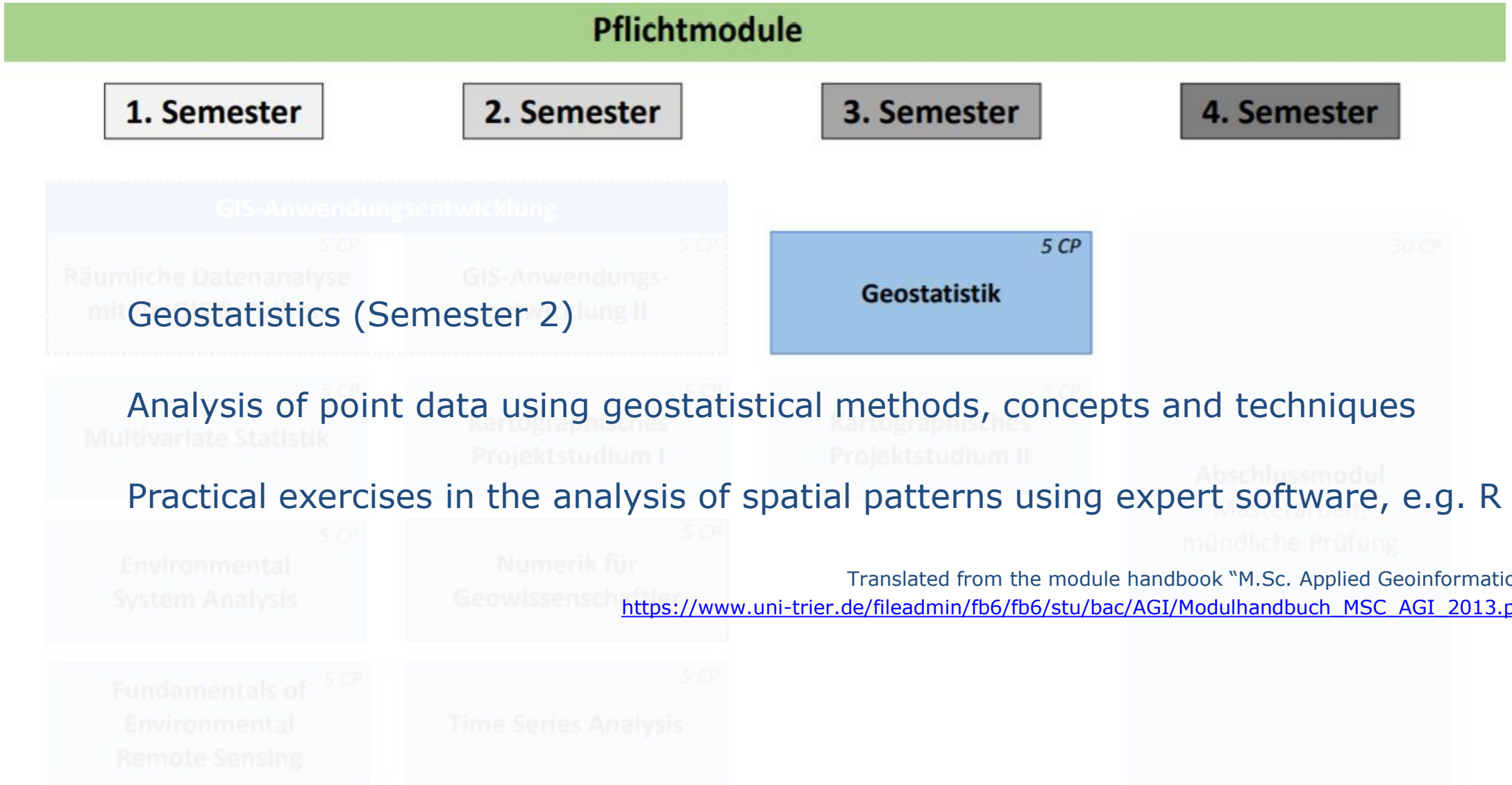
Clustering spatiotemporal point data to visualize spatial patterns

Dr. David Frantz

Geoinformatics – Spatial Data Science
Demonstration lecture
Trier / Zoom, 02.07.2020



https://www.uni-trier.de/fileadmin/fb6/fb6/stu/bac/AGI/Modulplan_MSc_AGI_Stand2016.pdf



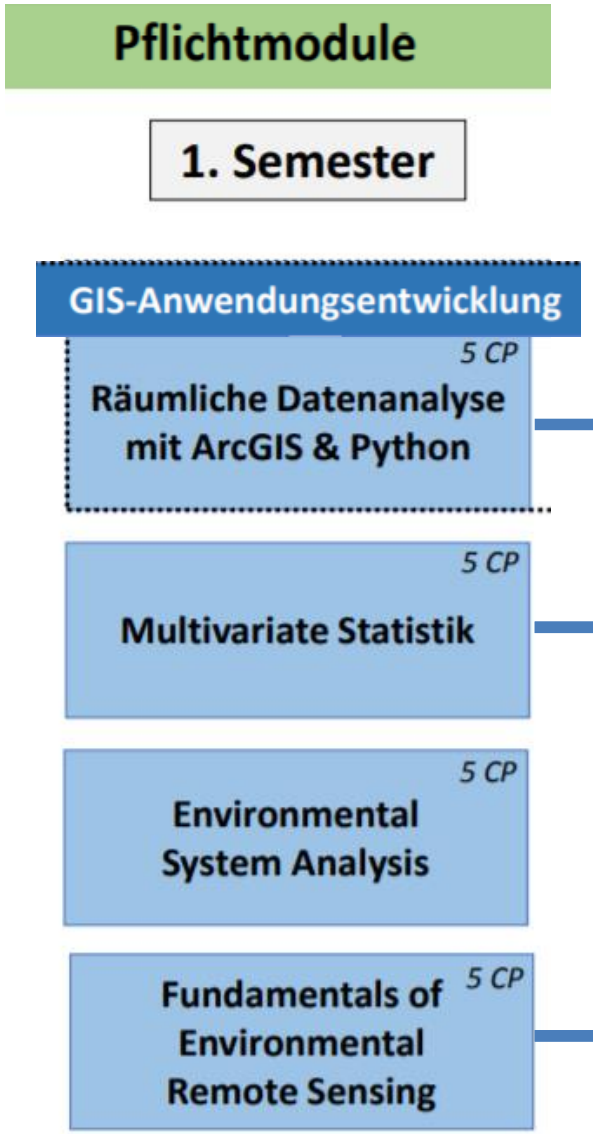
Geostatistics (Semester 2)

Analysis of point data using geostatistical methods, concepts and techniques

Practical exercises in the analysis of spatial patterns using expert software, e.g. R

Translated from the module handbook "M.Sc. Applied Geoinformatics"
https://www.uni-trier.de/fileadmin/fb6/fb6/stu/bac/AGI/Modulhandbuch_MSC_AGI_2013.pdf

Requirements



Processing of vector data, visualization of geodata
→ Knowledge of geospatial data analysis

Fundamental knowledge of relevant multivariate methods for predicting and testing, investigating dependencies and classification
→ Knowledge of statistics and concepts

Consolidation of skills in ~~SPSS/Matlab~~
-> some **R** skills

Cluster analysis
→ Knowledge of the concept

Translated from the module handbook "M.Sc. Applied Geoinformatics"
https://www.uni-trier.de/fileadmin/fb6/fb6/stu/bac/AGI/Modulhandbuch_MSC_AGI_2013.pdf

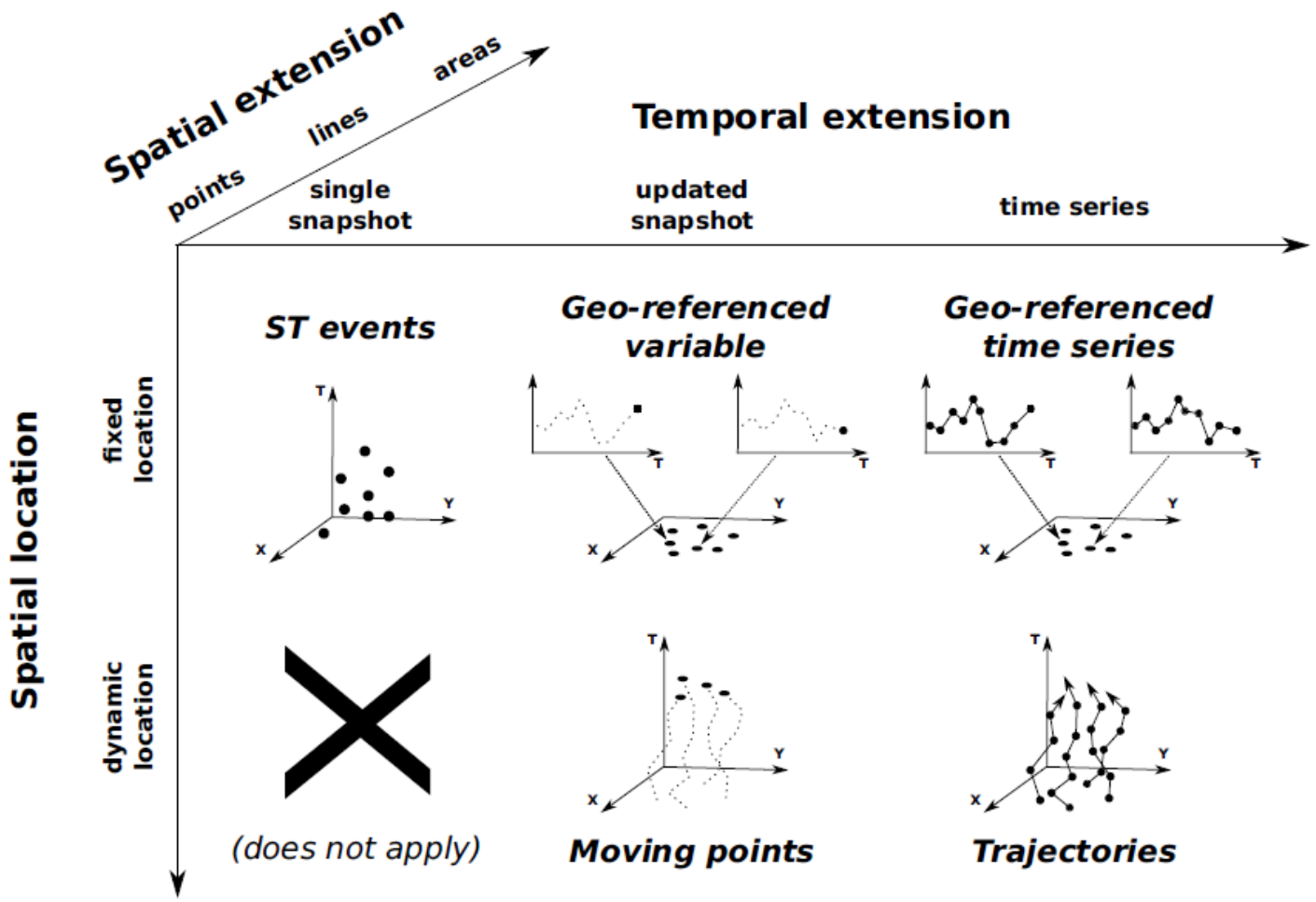
Learning Objective

- Introduction to spatiotemporal data
- Clustering algorithm
- Practical experience/demonstration to cluster real-life ST data with current relevancy

Spatiotemporal data

1) ST event

- Single measurement
- <longitude, latitude, timestamp>
  - seismic event
  - record of an epidemic



Kisilevich et al.: Spatio-temporal clustering. In: *Data mining and knowledge discovery handbook*. Springer, Boston, MA, 2009. S. 855-874.

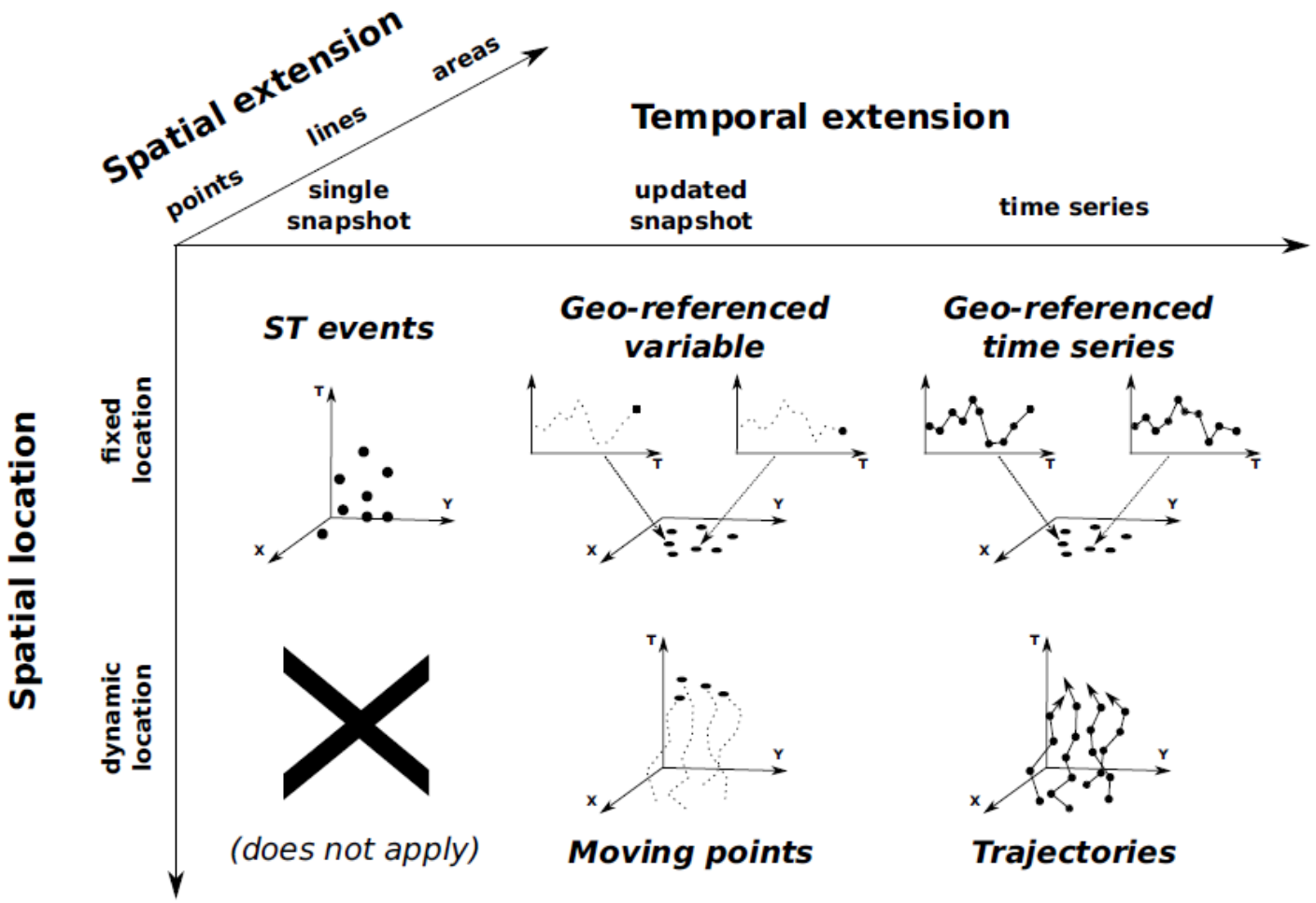
Spatiotemporal data

1) ST event

- Single measurement
- <longitude, latitude, timestamp>
  - seismic event
  - record of an epidemic

2) Geo-referenced variable

- Evolution in time, but only the most recent value
- <longitude, latitude, timestamp, non-spatial value>
  - Weather station with most recent temperature value

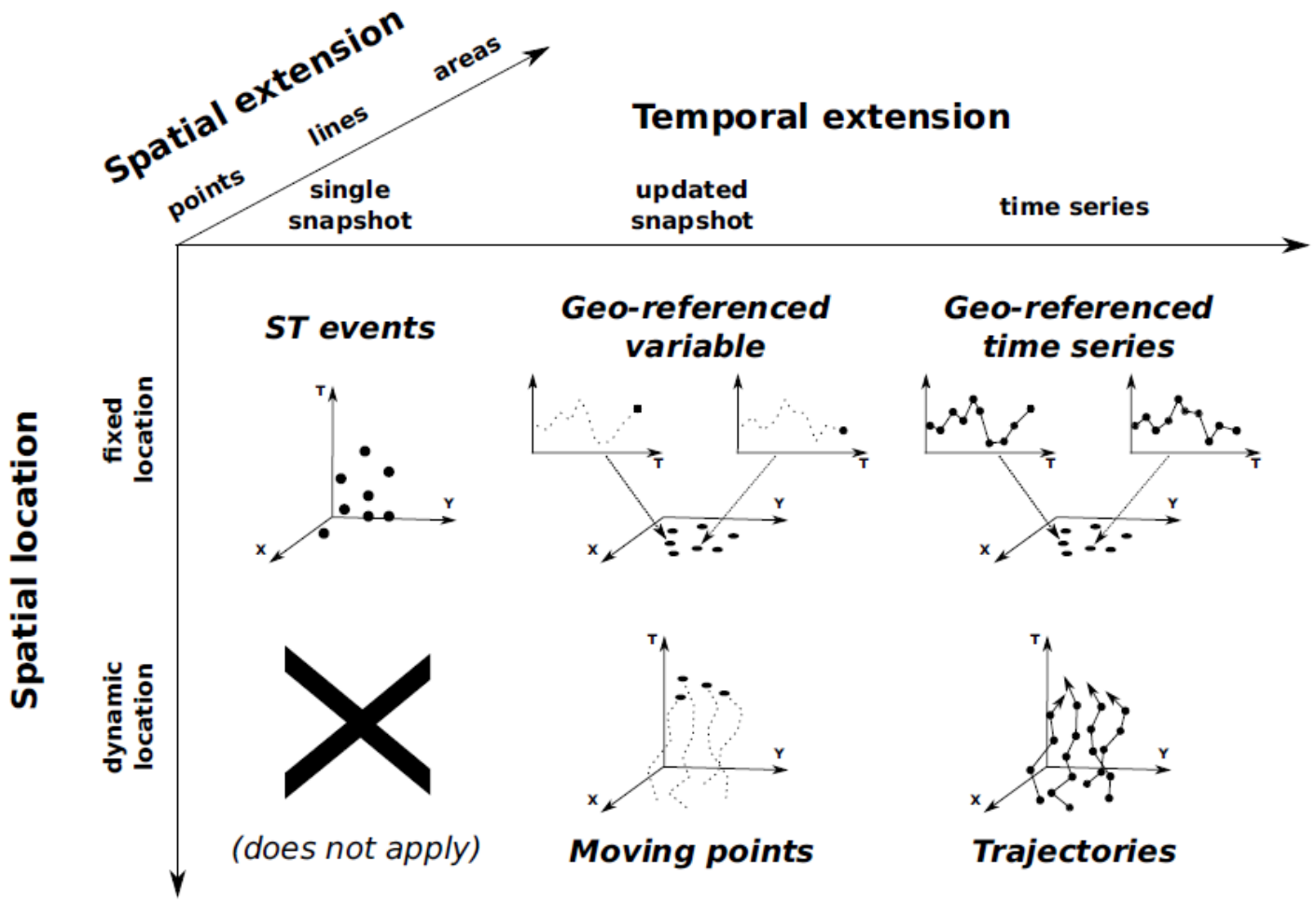


Kisilevich et al.: Spatio-temporal clustering. In: *Data mining and knowledge discovery handbook*. Springer, Boston, MA, 2009. S. 855-874.

Spatiotemporal data

3) Geo-referenced time series

- Whole history is stored
  - Climate station
  - NDVI time series



Kisilevich et al.: Spatio-temporal clustering. In: *Data mining and knowledge discovery handbook*. Springer, Boston, MA, 2009. S. 855-874.

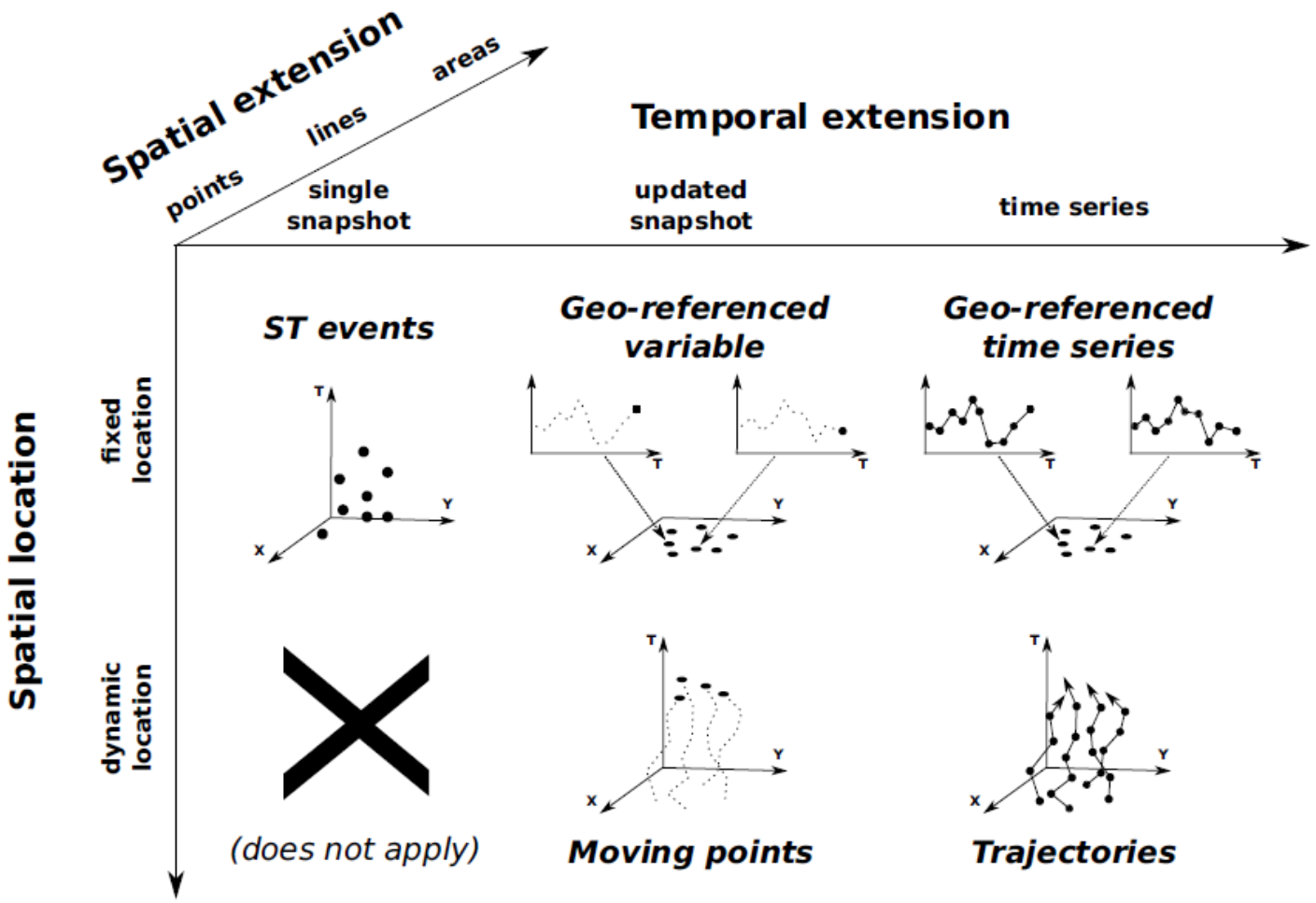
Spatiotemporal data

3) Geo-referenced time series

- Whole history is stored
  - Climate station
  - NDVI time series

4) Moving points

- object moves, most recent position
  - real-time tracking of vehicles



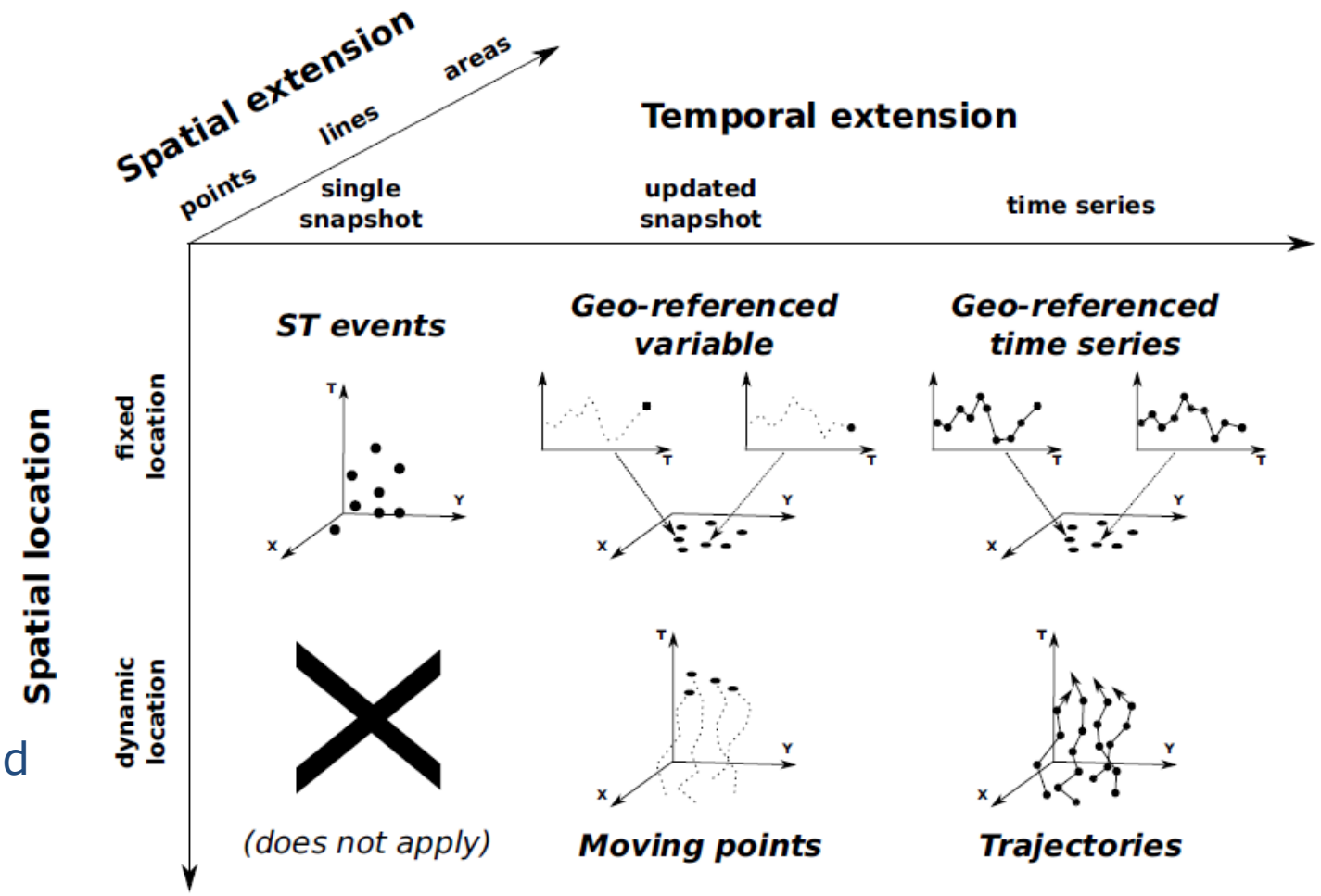
Kisilevich et al.: Spatio-temporal clustering. In: *Data mining and knowledge discovery handbook*. Springer, Boston, MA, 2009. S. 855-874.

Spatiotemporal data

3) Geo-referenced time series
- Whole history is stored
  - Climate station
  - NDVI time series

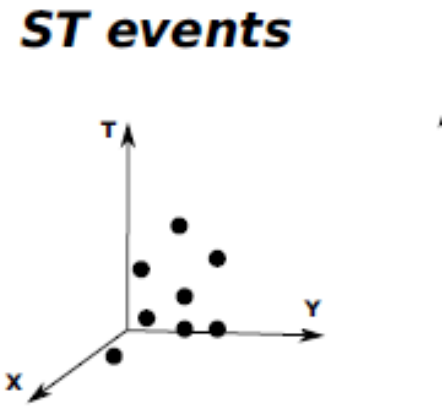
4) Moving points
- object moves, most recent position
  - real-time tracking of vehicles

5) Trajectories
- Object moves, whole history is stored
  - Google Location History



Kisilevich et al.: Spatio-temporal clustering. In: *Data mining and knowledge discovery handbook*. Springer, Boston, MA, 2009. S. 855-874.

Clustering ST event data



Three dimensions:
<longitude, latitude, timestamp>

Static in space and time = snapshot

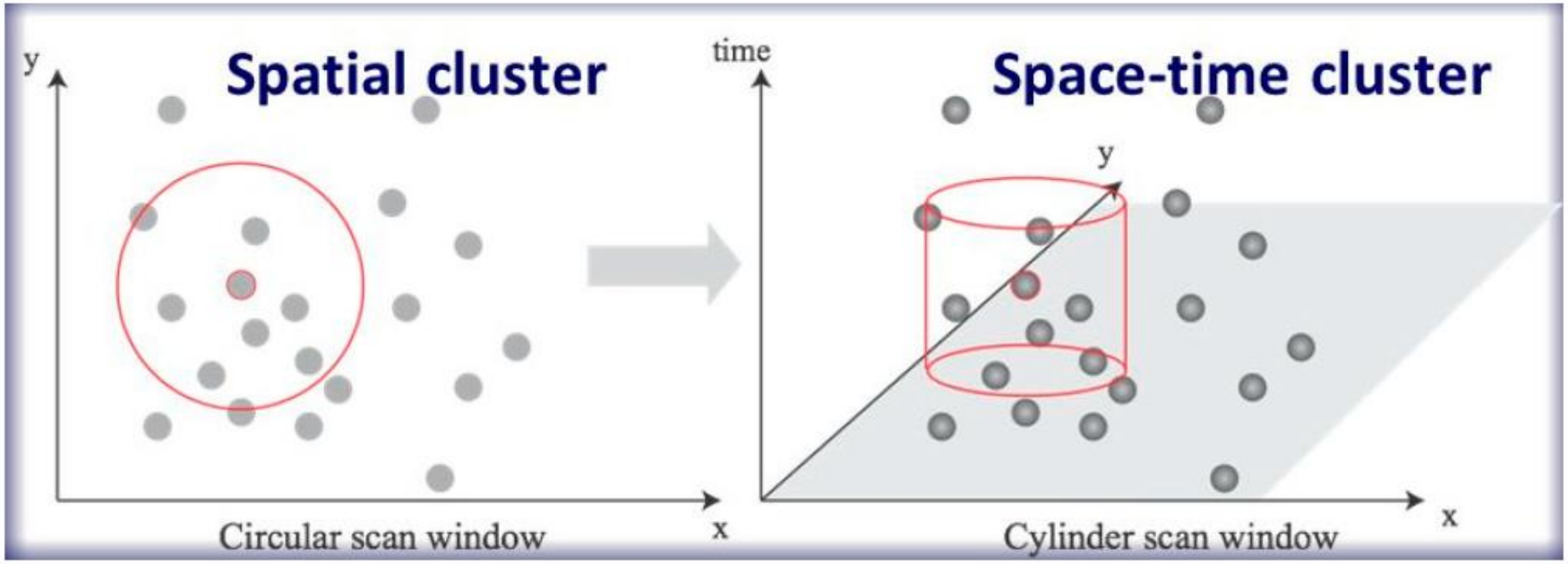
Problem: complex datasets
Solution: Spatiotemporal analyses methods to mine meaningful patterns for better understanding

Clustering = unsupervised method for discovering potential patterns

Finding clusters among events means to discover groups that lie close both in time and in space

Clustering ST event data

Classical example: spatio-temporal cylinders where the density of events is higher than outside



KULLDORFF, Martin. A spatial scan statistic. *Communications in Statistics-Theory and methods*, 1997, 26. Jg., Nr. 6, S. 1481-1496.

SHI, Zhicheng; PUN-CHENG, Lilian SC. Spatiotemporal data clustering: a survey of methods. *ISPRS international journal of geo-information*, 2019, 8. Jg., Nr. 3, S. 112.

DBSCAN

Density-Based Spatial Clustering of Applications with Noise

ESTER, Martin, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In: *Kdd*. 1996. S. 226-231.

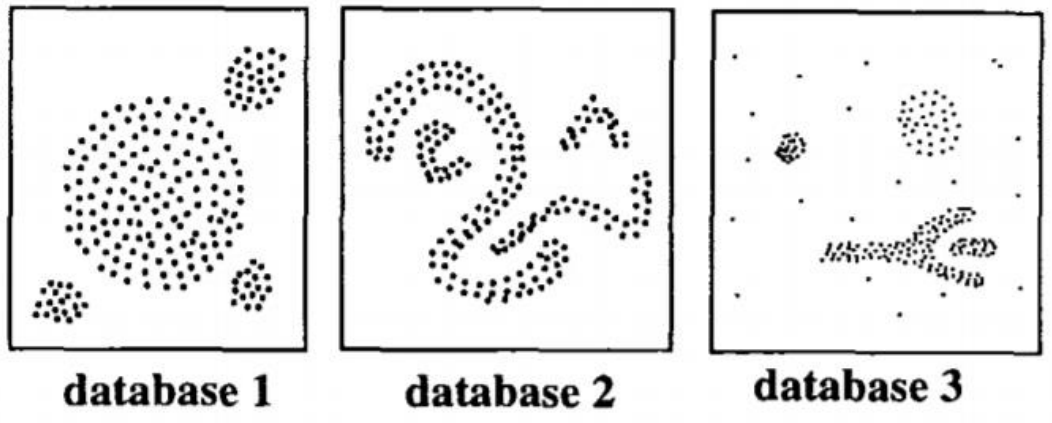
Density-based notion of cluster:
Within each cluster, we have a typical density of points, which is considerably higher than outside

Popular algorithm in data mining

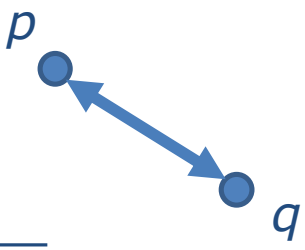
Find clusters of arbitrary shape
Detect noise

Only two input parameters, which
can be set using simple heuristics

Efficient, even for very large databases



DBSCAN concepts (1)

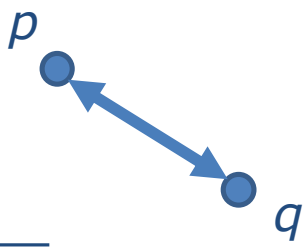


1) Neighborhood

Determined by a distance function, e.g. Euclidean Distance

Distance between two points p and q in database D: $dist(p,q) = \sqrt{(x_p - x_q)^2 + (y_p - y_q)^2}$

DBSCAN concepts (1)



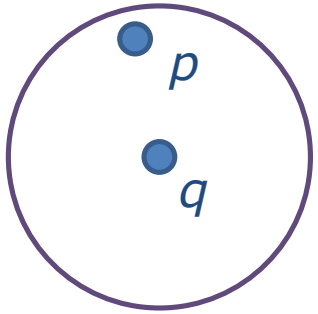
1) Neighborhood

Determined by a distance function, e.g. Euclidean Distance

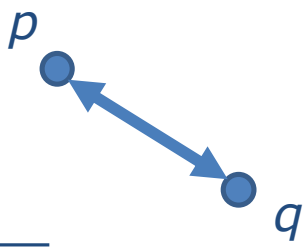
Distance between two points p and q in database D: $dist(p,q) = \sqrt{(x_p - x_q)^2 + (y_p - y_q)^2}$

2) Eps-neighborhood of a point q:

$N_{Eps}(q) = \{p \in D \mid dist(p,q) \leq Eps\}$



DBSCAN concepts (1)



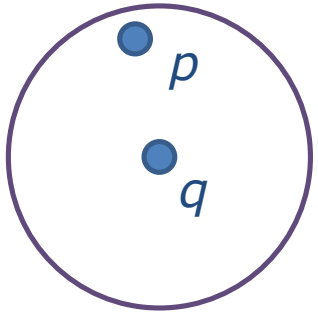
1) Neighborhood

Determined by a distance function, e.g. Euclidean Distance

Distance between two points *p* and *q* in database D: $dist(p,q) = \sqrt{(x_p - x_q)^2 + (y_p - y_q)^2}$

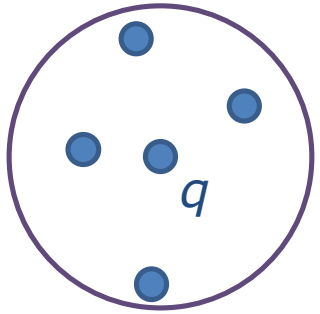
2) Eps-neighborhood of a point q:

$N_{Eps}(q) = \{p \in D \mid dist(p,q) \leq Eps\}$



3) Core point is part of a cluster *C*

$|N_{Eps}(q)| \geq MinPts$



MinPts = 3

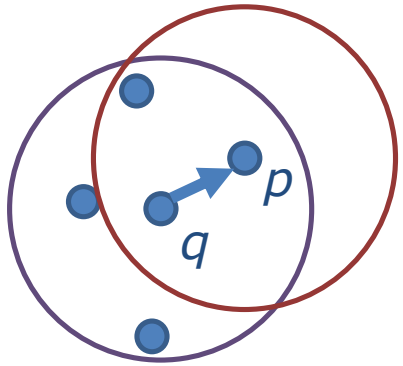
DBSCAN concepts (2)

4) Directly density-reachable

p is directly density-reachable from q if
p is within the Eps-neighborhood of q,
and q is a core point

$p \in N_{Eps}(q)$ AND

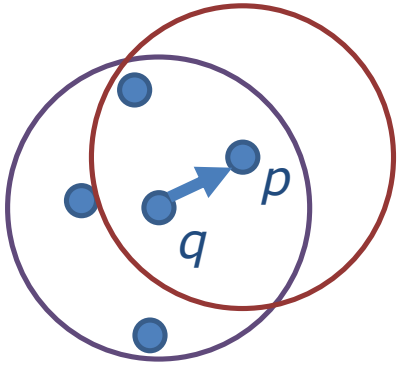
$\left|N_{Eps}(q)\right| \geq MinPts$



p directly density-reachable from q

q not directly density-reachable from p

DBSCAN concepts (2)



p directly density-reachable from q

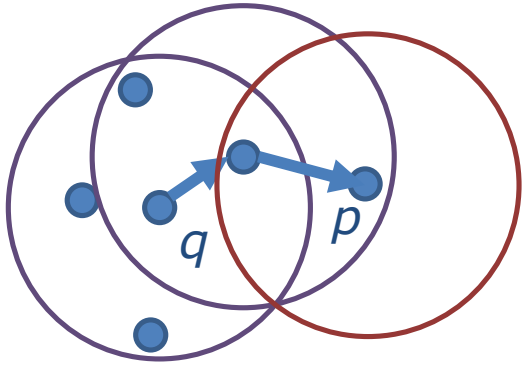
q not directly density-reachable from p

4) Directly density-reachable

p is directly density-reachable from *q* if
p is within the Eps-neighborhood of *q*,
and *q* is a core point

$p \in N_{Eps}(q)$ AND

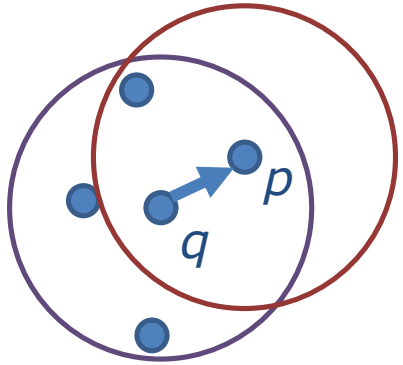
$\left|N_{Eps}(q)\right| \geq MinPts$



5) Density-reachable

p is density-reachable from *q* if there is a chain of points that are directly density-reachable

DBSCAN concepts (2)



p directly density-reachable from q

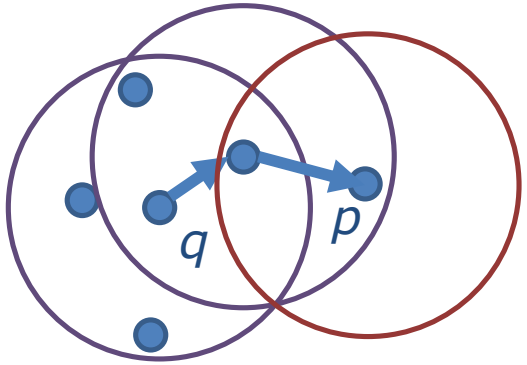
q not directly density-reachable from p

4) Directly density-reachable

p is directly density-reachable from *q* if *p* is within the Eps-neighborhood of *q*, and *q* is a core point

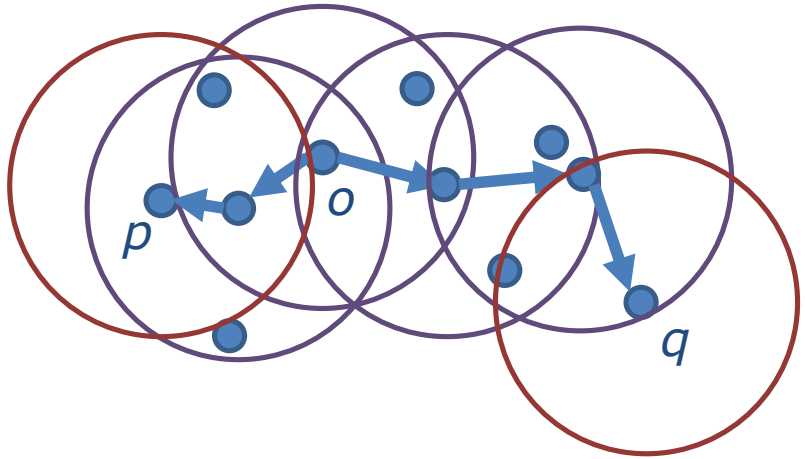
$p \in N_{Eps}(q)$ AND

$\left|N_{Eps}(q)\right| \geq MinPts$



5) Density-reachable

p is density-reachable from *q* if there is a chain of points that are directly density-reachable



6) Density-connected

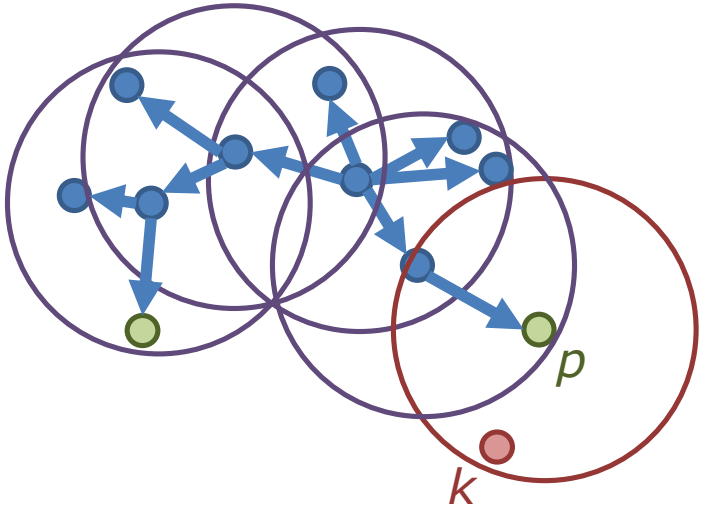
p is density connected to *q*, if both *p* and *q* are density-reachable from a point *o*

DBSCAN concepts (3)

7) Density-based cluster contains all points that are density-reachable from a seed point p:

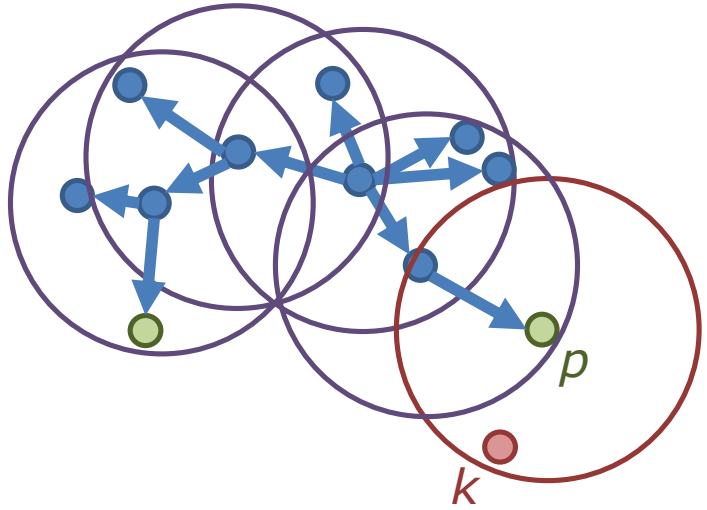
$\forall\, p, q: if\ p \in C$ AND q is density-reachable from p

$\forall\, p, q \in C: if\ p$ is density-connected to q



DBSCAN concepts (3)

7) Density-based cluster contains all points that are density-reachable from a seed point p:
$\forall\, p, q: if\ p \in C$ AND q is density-reachable from p
$\forall\, p, q \in C: if\ p$ is density-connected to q



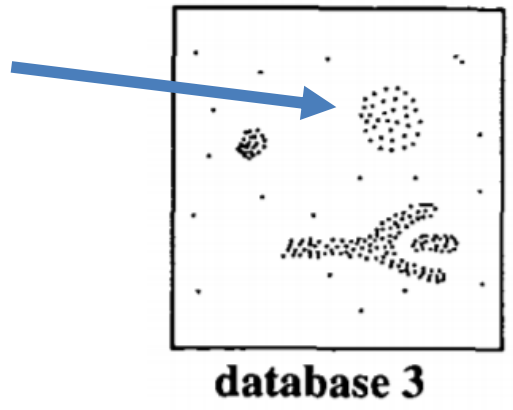
8) Border point
p is a border point if not a core point, but density reachable from another core point

9) Noise
Any point k not belonging to any cluster

Eps and MinPts

MinPts does not critically affect clustering results
It is suggested to use 4, or the number of dimensions + 1, denoted as *k*

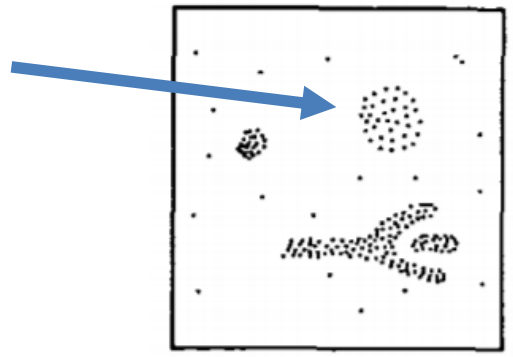
The distance *Eps* should be set according to the "thinnest" cluster



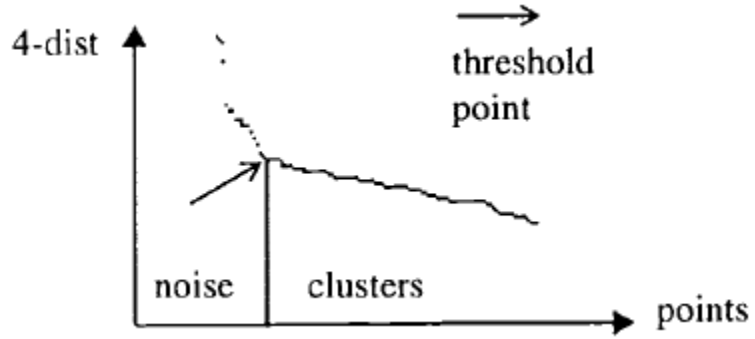
Eps and MinPts

MinPts does not critically affect clustering results
It is suggested to use 4, or the number of dimensions + 1, denoted as *k*

The distance *Eps* should be set according to the "thinnest" cluster



Simple solution:
Compute the distance d of a point p to its k-th nearest neighbor
Repeat for each point
Sort the distances and plot (*k-dist graph)*



Spatiotemporal clustering

Main challenge: different units of time and space..

What distance equals a unit of time?

Depends on the application:

- Two pedestrians that have met with a distance of 1.5m within a minute interval could belong to the same cluster (e.g. CORONA App)
- Two spatially close sample points in a physics experiment that are nanoseconds apart could belong to different clusters

DBSCAN for spatiotemporal data

1) ST-DBSCAN

- Extension of DBSCAN, which can handle ST data (or other non-ST variables)
  - requires another distance parameter for the temporal domain (*Eps2*),
  - both *Eps1* and *Eps2* can be derived from *k-dist graphs*

BIRANT, Derya; KUT, Alp. ST-DBSCAN: An algorithm for clustering spatial–temporal data. *Data & knowledge engineering*, 2007, 60. Jg., Nr. 1, S. 208-221.

DBSCAN for spatiotemporal data

1) ST-DBSCAN

- Extension of DBSCAN, which can handle ST data (or other non-ST variables)
  - requires another distance parameter for the temporal domain (*Eps2*),
  - both *Eps1* and *Eps2* can be derived from *k-dist graphs*

BIRANT, Derya; KUT, Alp. ST-DBSCAN: An algorithm for clustering spatial–temporal data. *Data & knowledge engineering*, 2007, 60. Jg., Nr. 1, S. 208-221.

2) Classical DBSCAN

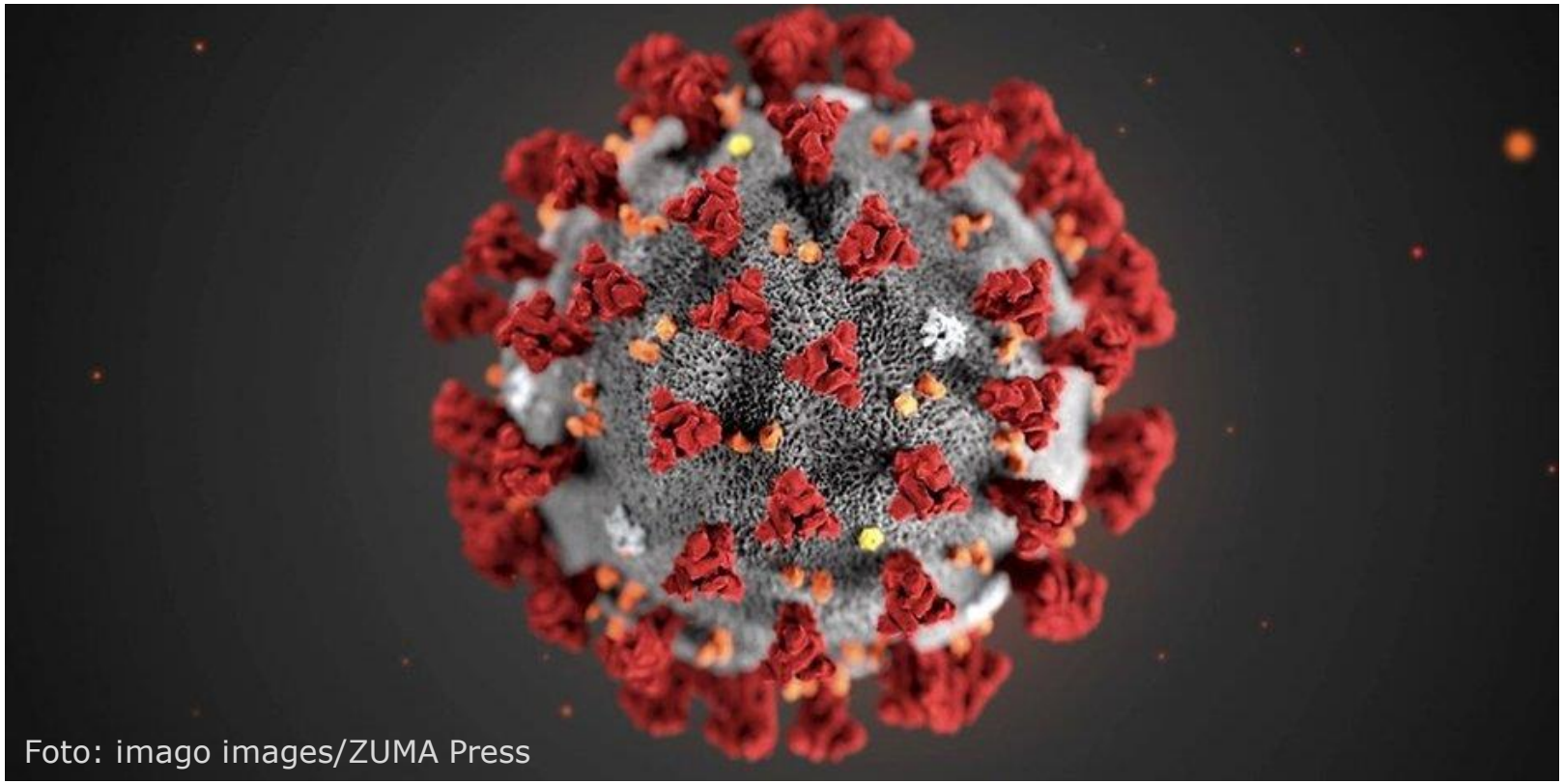
- Can be applied to 2D, 3D or any Euclidean high dimensional feature space
  - Temporal dimension is simply an additional dimension:

$$dist(p,q) = \sqrt{(x_p - x_q)^2 + (y_p - y_q)^2 + (t_p - t_q)^2}$$

→ some sort of scaling might be required to use the same *Eps* for space AND time

Hands-on session

COVID-19 (Coronavirus SARS-CoV-2)



Play with the data

Download the JupyterLab environment from

github.com/davidfrantz/covid19

includes

- Jupyter notebooks with all plots and code,
- COVID-19 data,
- this presentation,
- literature with suggested reading

requires

- JupyterLab
- R & R-Kernel

Parameters that will affect the clusters

- Number of cases N
  → find larger or smaller hotspots,
  → incidence instead of abs. numbers?

- Scaling of the temporal dimension
  → 7 days, 31 days?
  → statistical rescaling method for all dimensions? (e.g. z-transform)

- Eps
  → Shift the allocations to noise/clusters

Stay healthy. Don’t become a cluster!